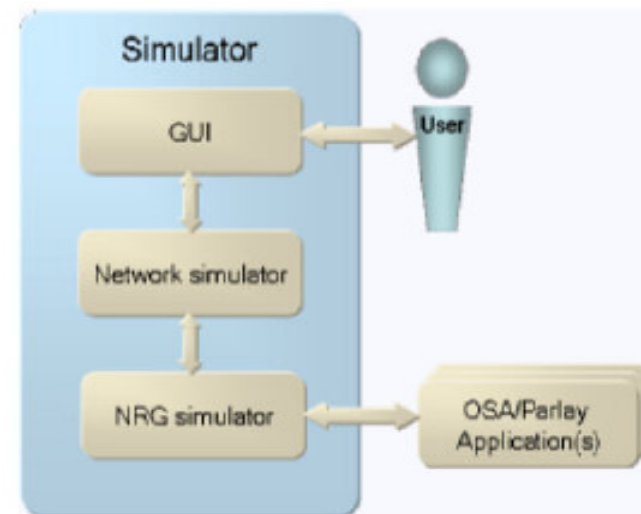
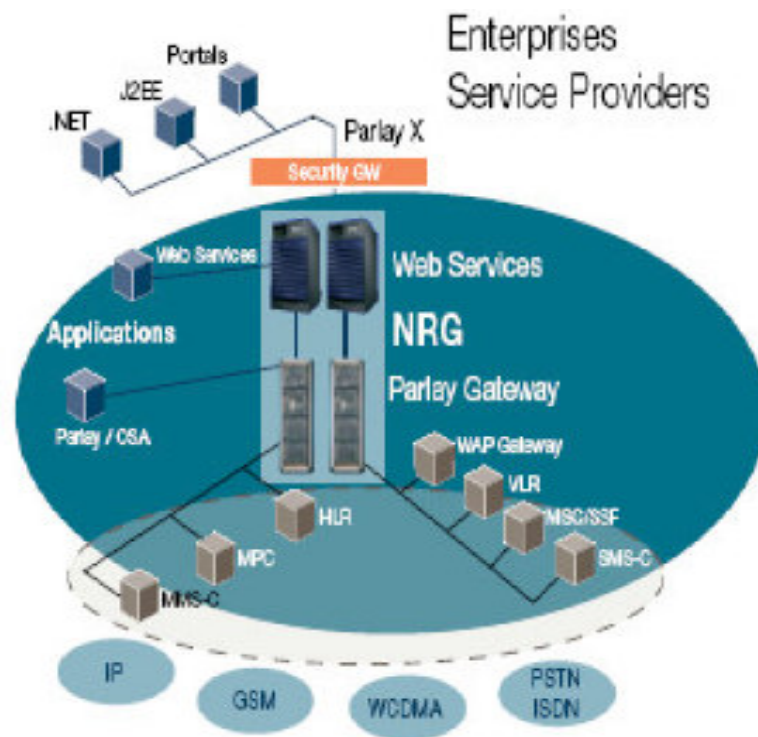


# Laboratório 1: Aplicações em Parlay/OSA

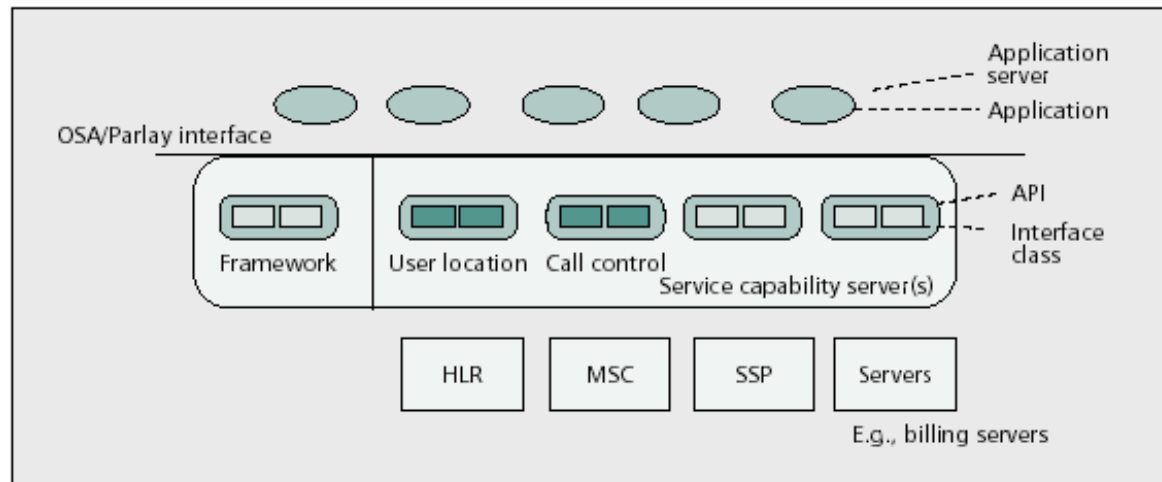
- Ambientação com a plataforma NRG
- Aprendizagem conceitos Parlay/OSA
- Representação formal das aplicações



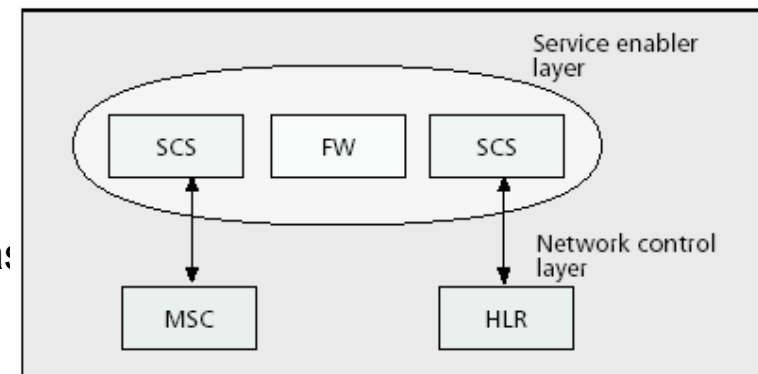
# 0: Configuração da Máquina Virtual

- Username:IGRS11-t1-0X Password: P@assw0rd
- Máquina virtual na directoria IGRS-2010-Lab1, copiar para Documents
- Editar `Windows XP Professional.vmx`  
alterar o endereço MAC para:  
`ethernet0.address = "00:50:56:00:54:1X"`

# Entidades lógicas Parlay/OSA



- SCS implementam a parte servidor e as Aplicações a parte cliente das APIs
- SCS interagem com os elementos chave da rede (e.g., SSP, HLR)
- A *Framework* controla e medeia o acesso das aplicações aos serviços
- OSA acrescenta ao Parlay recomendações para o mapeamento das interfaces nos protocolos de acesso à rede



# As interfaces Parlay/OSA

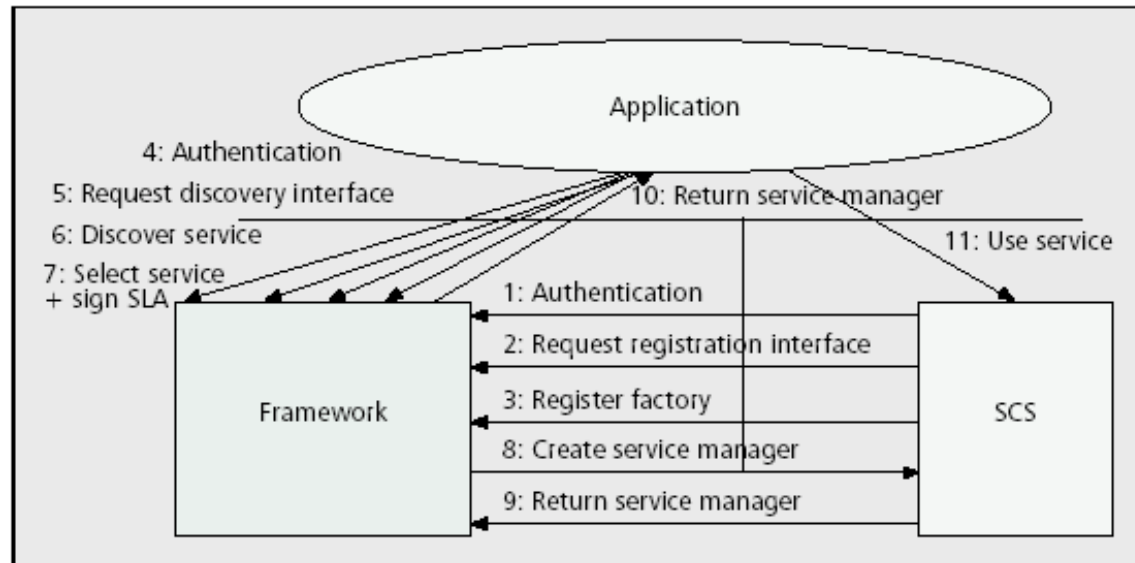
API	Descrição
Framework	Segurança, integridade, autenticação
Call Control	Estabelecimento, fecho e gestão de chamadas, conferências e chamadas multimédia. Notificações associadas a chamadas ou ligações. Diferentes especializações (Generic Call Control, Multiparty Call Control)
Data Session Control	Estabelecimento, fecho e gestão de sessões de dados (ex., transferência de ficheiros)
User Interaction	Interacção com o <i>end-user</i> para tocar ou apresentar mensagens e/ou recolher informação introduzida pelo <i>end-user</i>
Mobility	Obtenção de informação sobre a localização e/ou estado do <i>end-user</i>
Generic Messaging	Envio e recepção de mensagens (email, voz, SMS, MMS). Gestão de caixas de correio, voz
Terminal Capabilities	Obtenção de informação sobre as capacidades de um terminal (ex., tipo e resolução do ecrã)
Connactivity Management	Negociação e gestão da qualidade de serviço (QoS) e contratos de nível de serviço (SLA) em redes IP
Account Mgmt.	Criação, remoção e gestão de contas de assinantes
Charging	Reserva ou cobrança de fundos de acordo com a conta de um assinante

# As interfaces Parlay/OSA

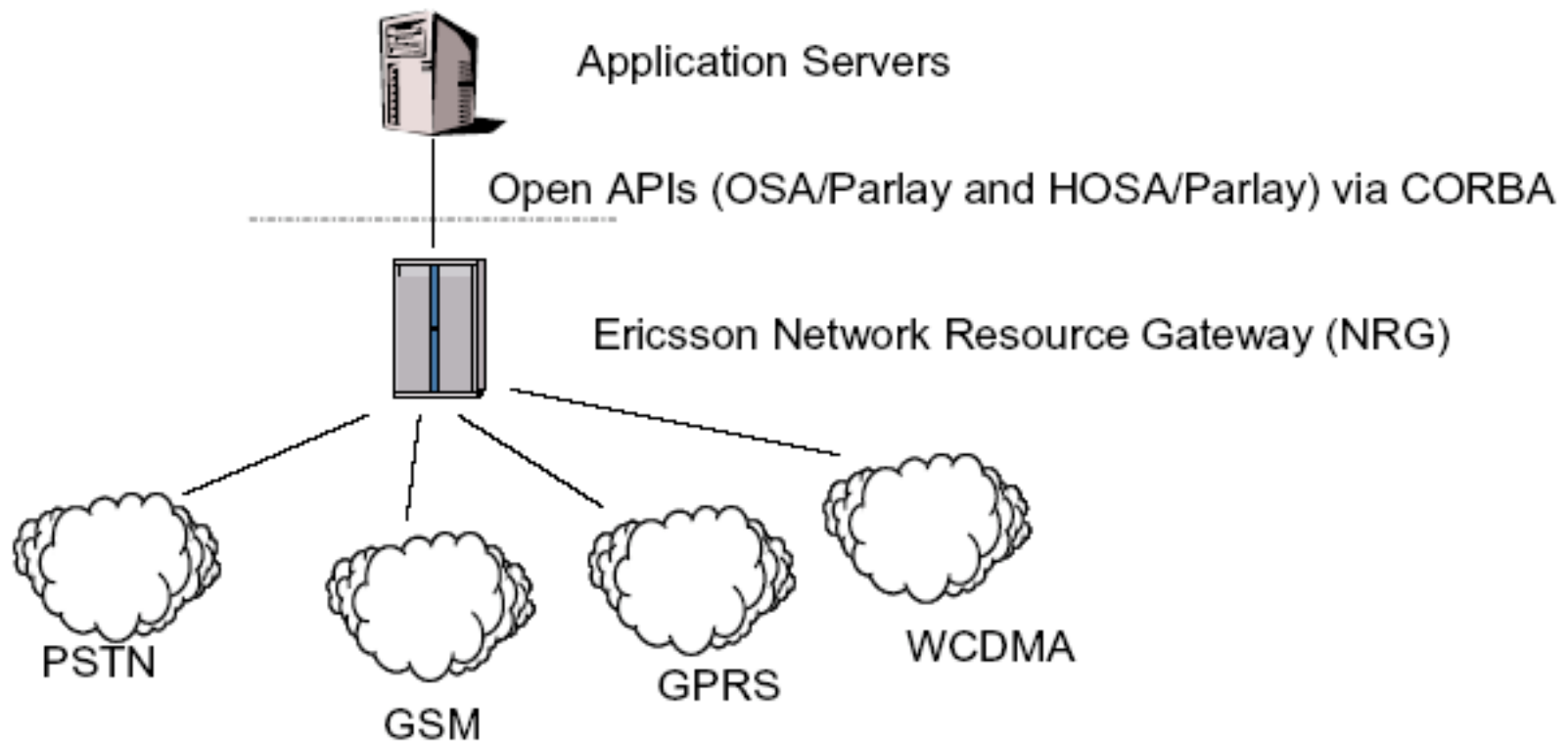
API	Descrição
Framework	Segurança, integridade, autenticação
Call Control	Estabelecimento, fecho e gestão de chamadas, conferências e chamadas multimédia. Notificações associadas a chamadas ou ligações. Diferentes especializações (Generic Call Control, Multiparty Call Control)
Data Session Control	Estabelecimento, fecho e gestão de sessões de dados (ex., transferência de ficheiros)
User Interaction	Interacção com o <i>end-user</i> para tocar ou apresentar mensagens e/ou recolher informação introduzida pelo <i>end-user</i>
Mobility	Obtenção de informação sobre a localização e/ou estado do <i>end-user</i>
Generic Messaging	Envio e recepção de mensagens (email, voz, SMS, MMS). Gestão de caixas de correio, voz
Terminal Capabilities	Obtenção de informação sobre as capacidades de um terminal (ex., tipo e resolução do ecrã)
Connactivity Management	Negociação e gestão da qualidade de serviço (QoS) e contratos de nível de serviço (SLA) em redes IP
Account Mgmt.	Criação, remoção e gestão de contas de assinantes
Charging	Reserva ou cobrança de fundos de acordo com a conta de um assinante

# A API Framework

- Inspirado na arquitectura TINA
- Acesso controlado as APIs
  - Autenticação e segurança
  - Descoberta de aplicações
  - Inclusão de múltiplos vendedores e APIs adicionais não normalizadas
  - Núcleo central de interfaces:
    - *Trust and security management*: autenticação de domínios
    - *Registration*: registo de novos SCSs
    - *Service life cycle mananagement* e *Service discovery* criação de novas API e descoberta de API
    - *Contract management*: gestão de contratos entre os diferentes participantes
- 1-3: Registo do SCS com a Framework
- 4: Autenticação do domínio da aplicação e verificação do contrato de serviço
- 5-6: Descoberta do serviço
- 7: Estabelecimento do contrato do tipo SLA
- 8-9: Criação de uma instância do serviço
- 10-11: Utilização do serviço

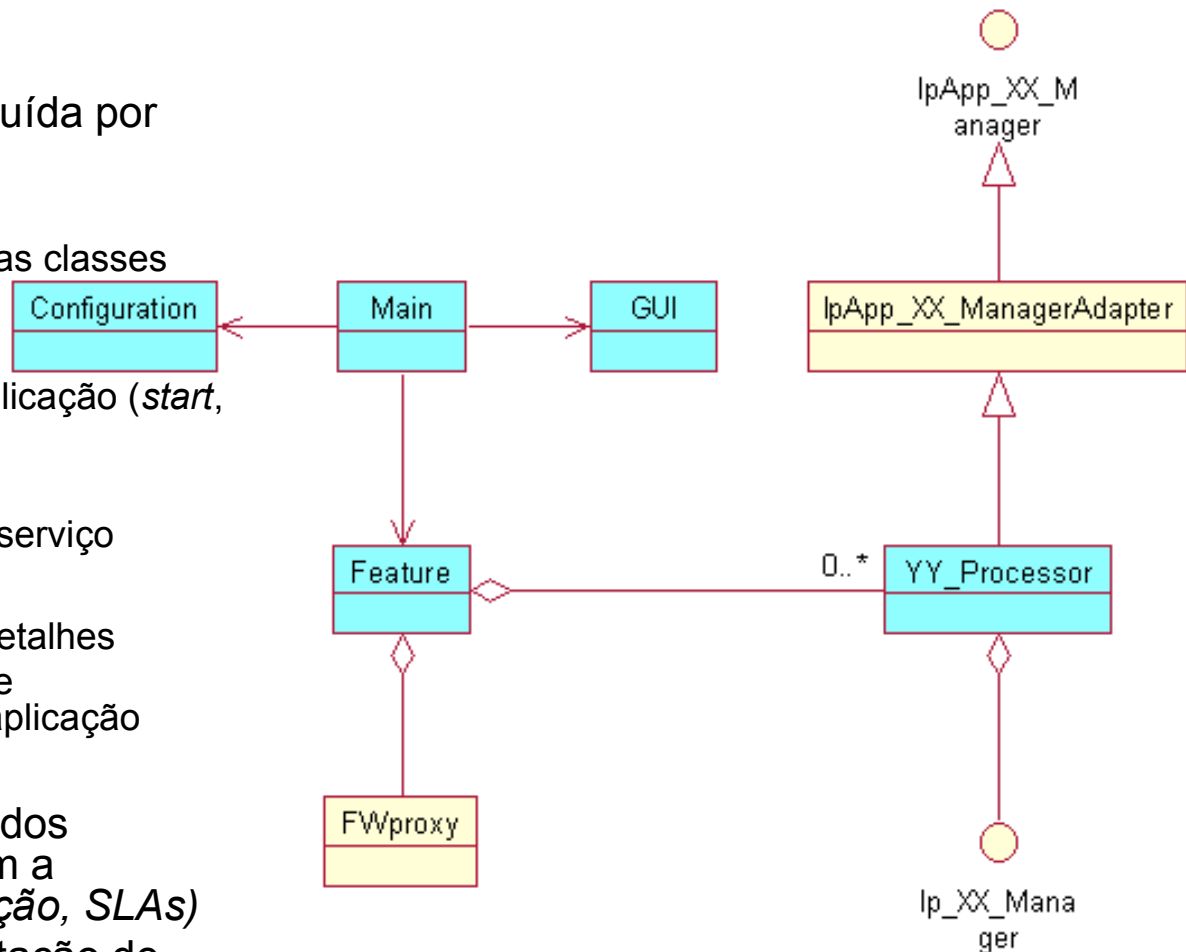


# O Network Resource Gateway (NRG) da Ericsson



# Diagrama de classes das aplicações exemplo do SDK da Ericsson

- Estrutura geral
  - Lógica da aplicação distribuída por várias classes
- Classe Main
  - Responsável por lançar as outras classes (interacção com o FWproxy)
- Classe GUI
  - Interacção com o *gestor* da aplicação (*start*, *stop*, acções)
- Classe Feature
  - Onde se encontra a lógica do serviço
- Classes YY\_Processor
  - Abstraem da classe *Feature* detalhes
  - Pode haver mais de um tipo de YY\_Processor numa mesma aplicação
- Classe FWproxy
  - Classe que abstrai alguns dos detalhes da interacção com a *Framework* (ex., *autenticação*, *SLAs*)
  - Permite a obtenção e libertação de *service managers*



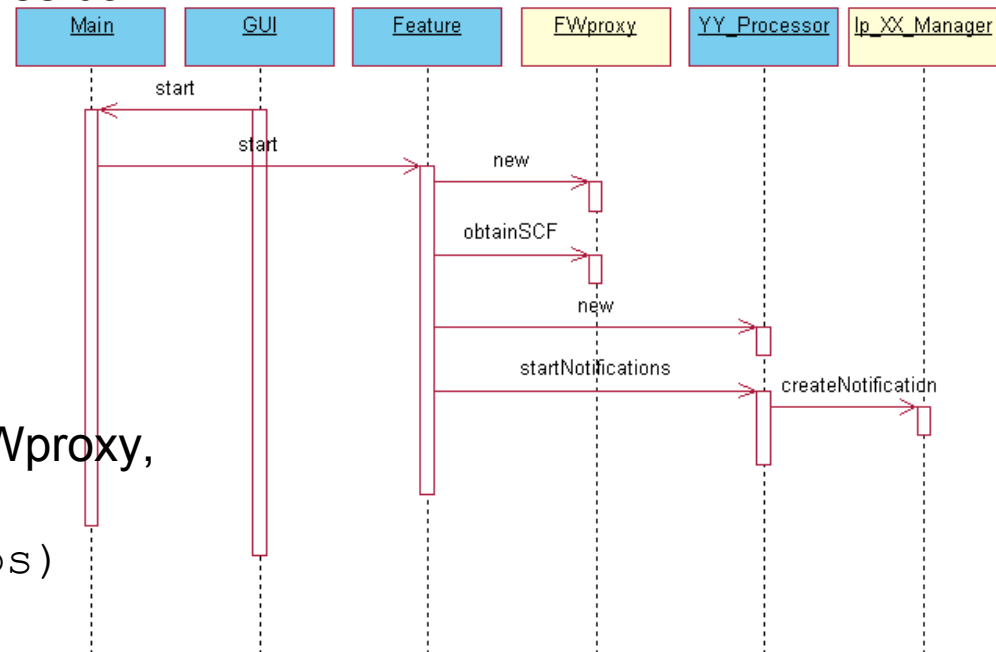


# A classe *Main* das aplicações exemplo

- Classe que abstrai alguns dos detalhes da interacção com a *Framework* (ex., autenticação, SLAs)

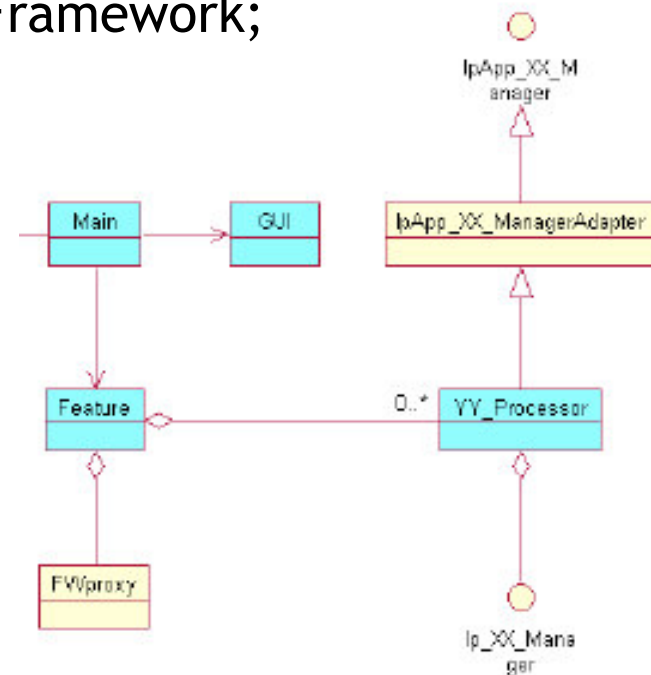
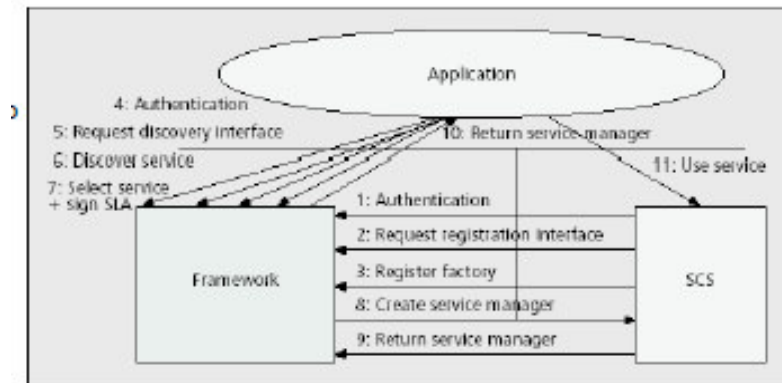
- Permite a obtenção e libertação de *service managers*

- Alguns métodos chave
  - // Criação de uma instância de FWproxy, interacção com a *Framework*  
**FWproxy** (Properties aProps)
  - // Obter um *service manager*  
public IService  
**obtainSCF** (String aSCFName)
  - // Libertar recursos  
void **releaseSCF** (IService aSCF)
  - // Libertar os recursos da *Framework* usados  
public void **endAccess** ()



## 1.1. Aplicação NullCallApp (21)

- Estudar e verificar experimentalmente:
  - Interação da Aplicação com a Framework Parlay:
  - Autenticação da aplicação com a Framework;
  - Negociar e aceitação de SLAs



- Utilização de SCS (não é feita nesta “Aplicação”)

# Diagrama de sequência para a interface *CallControl*

- Classe IpAppLogic

- Onde se encontra a lógica do serviço
- Faz pedidos

Criação e libertação da chamada (2,10)

Estabelecimento de ligações (5,8)

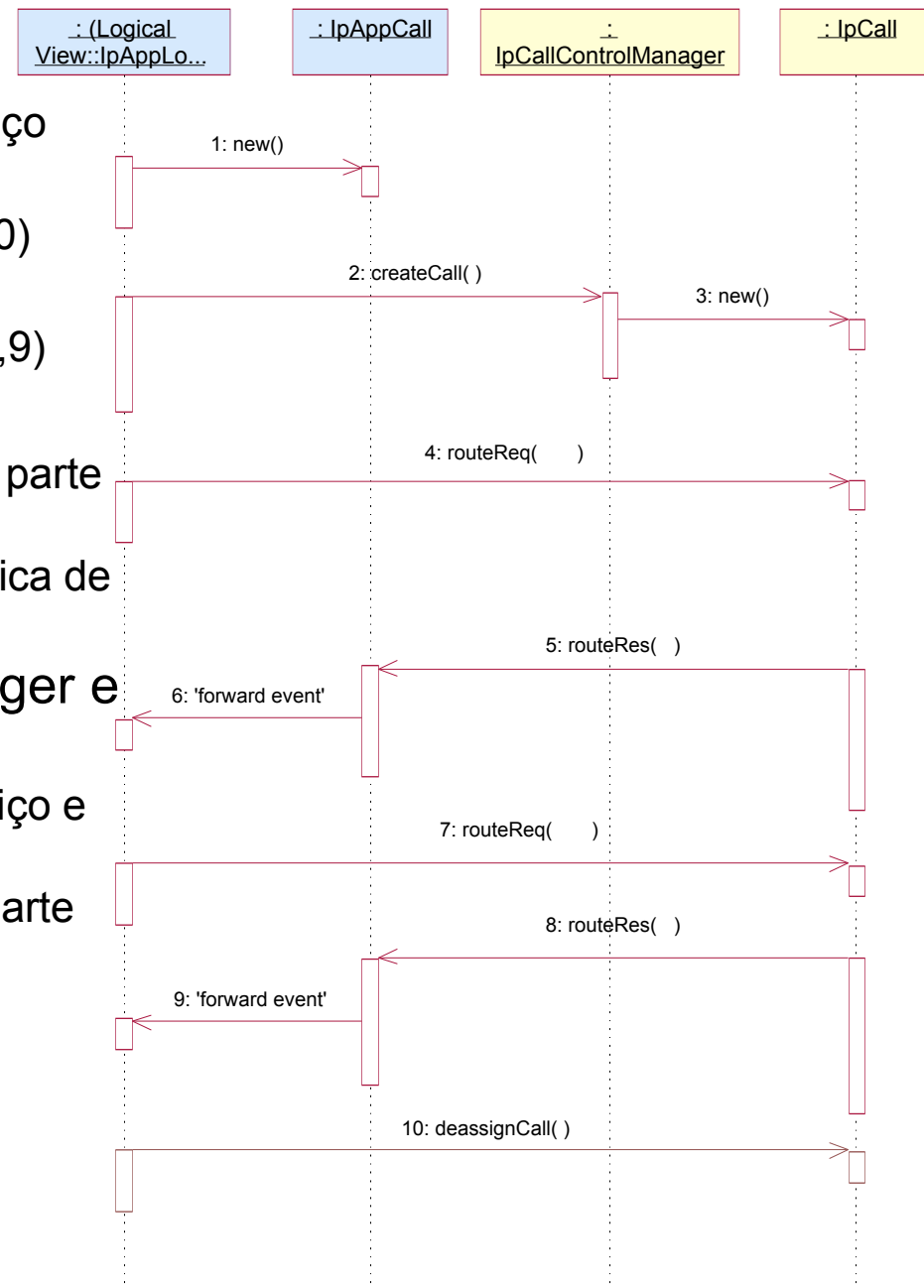
- Recebe notificações de eventos (6,9)

- Classe IpAppCall

- Recebe notificações/resultados da parte rede da interface (IpCall)
- Envia notificações/resultados à lógica de serviço

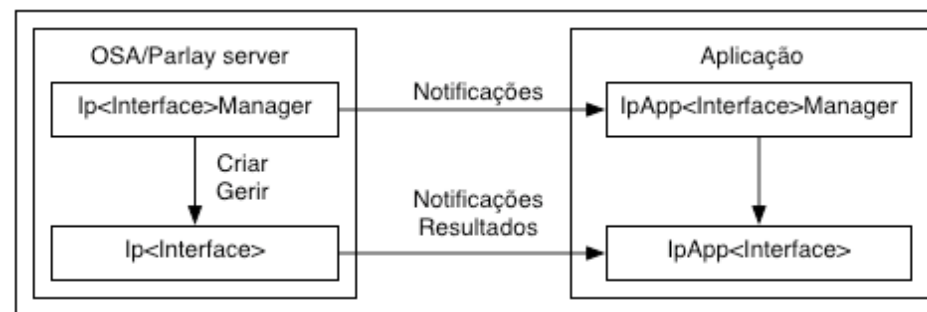
- Classes IpAppCallControlManager e IpCall

- Recebe, pedidos da lógica de serviço e da parte aplicação das interfaces
- Envia notificações/resultados à parte aplicação das interfaces

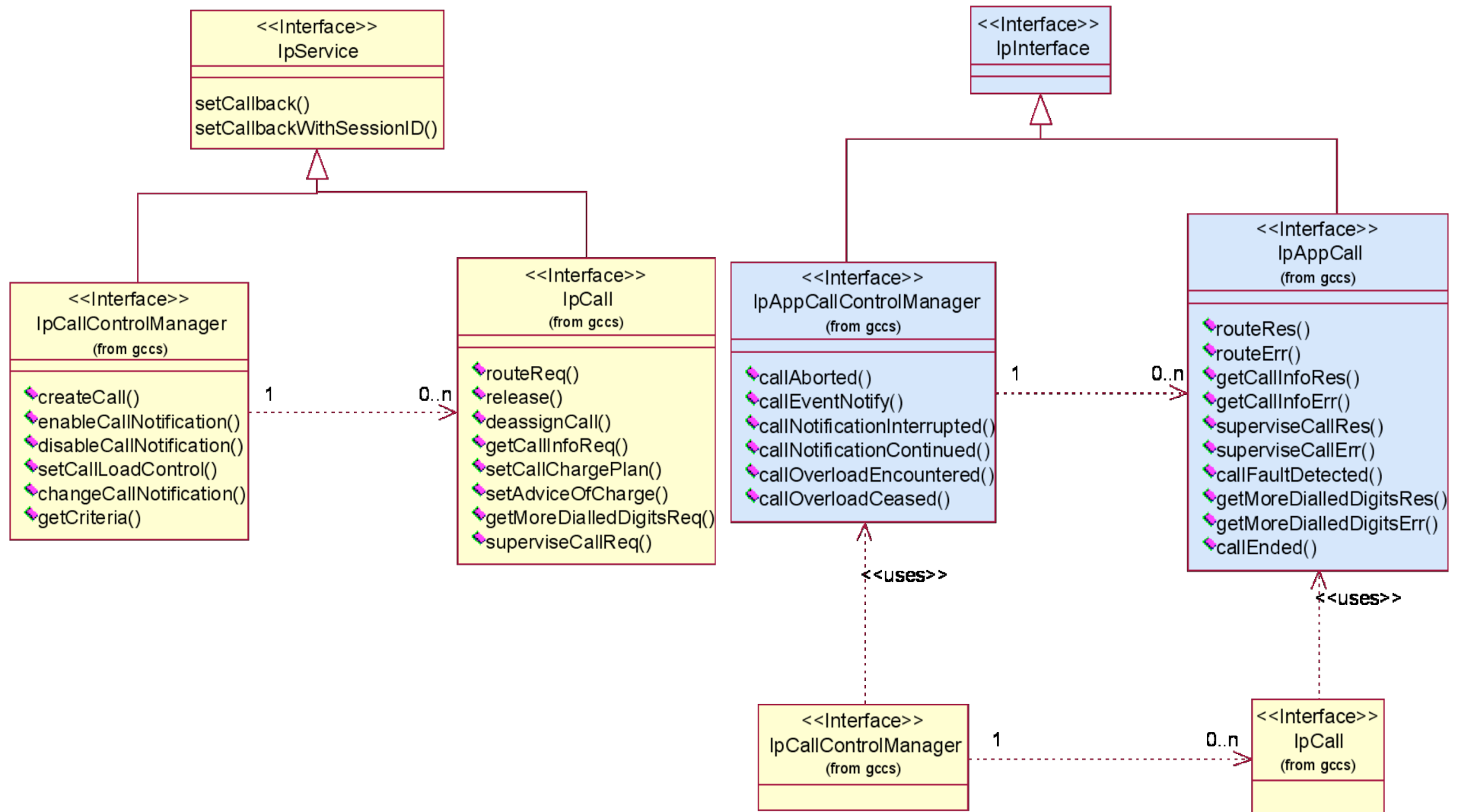


# As interfaces Parlay/OSA

- Especificadas e descritas utilizando a orientação aos objectos
  - Normativo
    - Especificação de interfaces (IDL) e e definições de tipos de dados (IDL)
  - Informativo
    - Diagramas de classes (UML) e diagramas de sequencias de estados (UML)
- Estrutura genérica de uma interface
  - Implementação da interface em duas partes (rede e aplicação)
  - Do lado da rede
    - Ip<Interface> operações para controlar os recursos da rede
    - Ip<Interface>Manager criação e gestão de intâncias de Ip<Interface>. Pedidos de notificações de eventos no servidor da aplicação
  - Do lado da aplicação (analogia *callback* interfaces)
    - IpApp<Interface> recepção resultados e notificações da parte rede da interface
    - IpApp<Interface>Manager criação/gestão de intâncias de IpApp<Interface>



# Diagrama de classes da interface *CallControl*

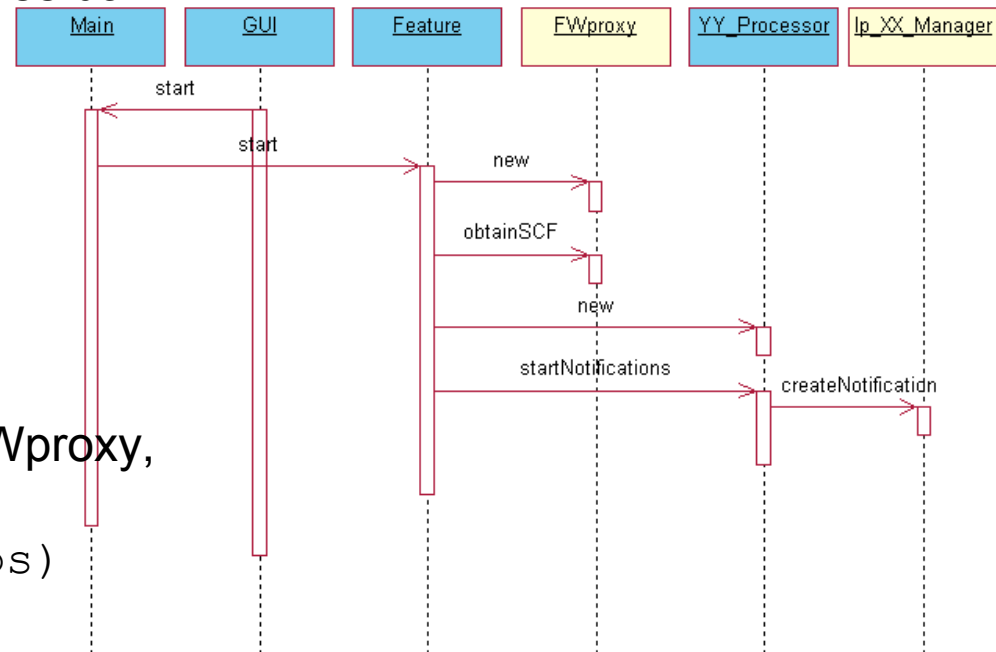


# A classe *Main* das aplicações exemplo

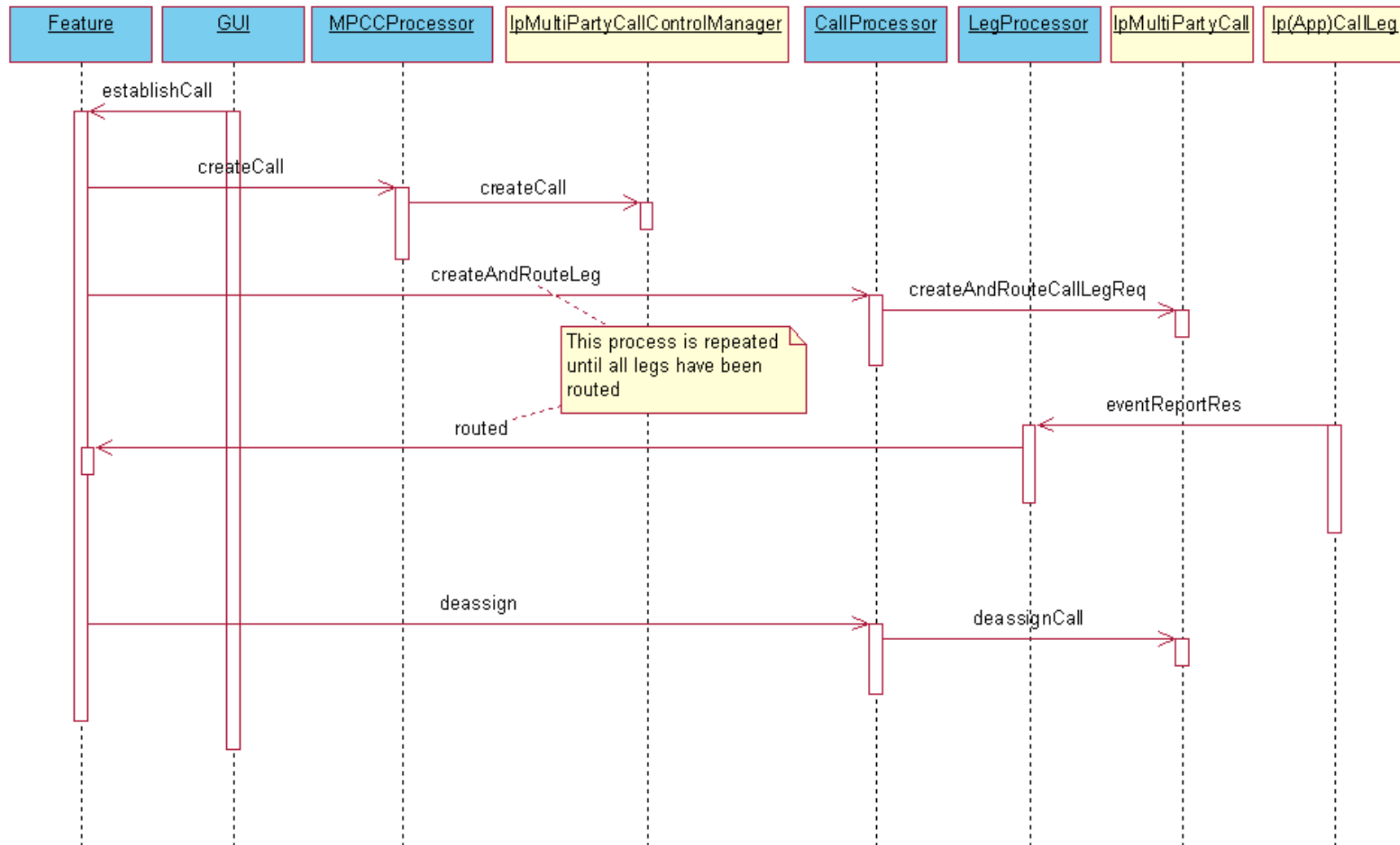
- Classe que abstrai alguns dos detalhes da interacção com a *Framework* (ex., autenticação, SLAs)

- Permite a obtenção e libertação de *service managers*

- Alguns métodos chave
  - // Criação de uma instância de FWproxy, interacção com a *Framework*  
**FWproxy** (Properties aProps)
  - // Obter um *service manager*  
public IService  
**obtainSCF** (String aSCFName)
  - // Libertar recursos  
void **releaseSCF** (IService aSCF)
  - // Libertar os recursos da *Framework* usados  
public void **endAccess** ()

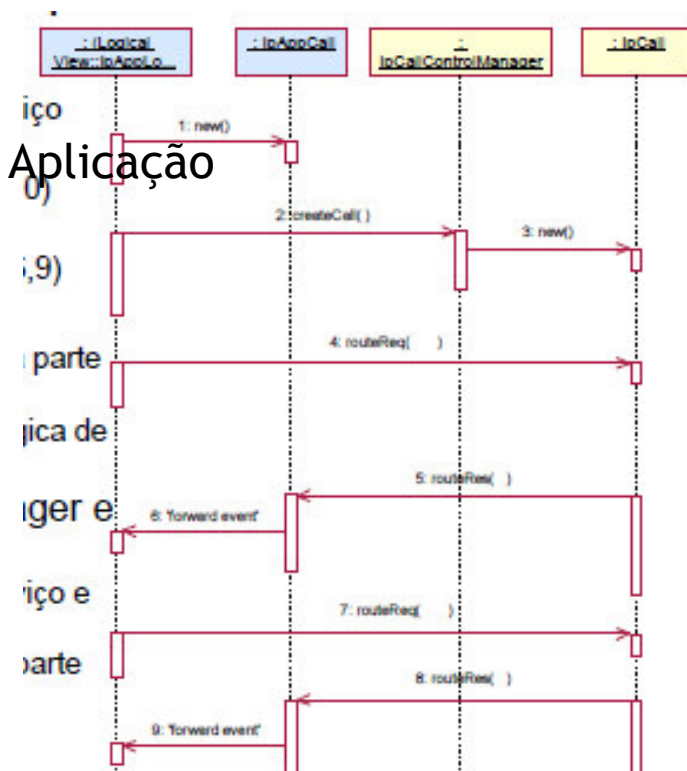
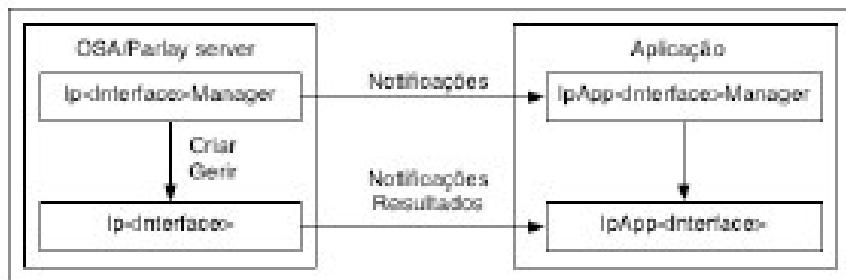


# Diagrama de sequência da aplicação exemplo *WebDial* do SDK da Ericsson



## 1.2. Aplicação CallMonitor (22)

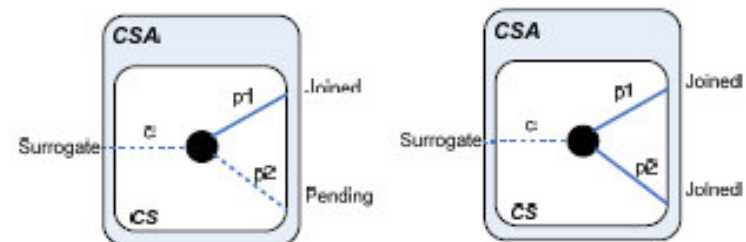
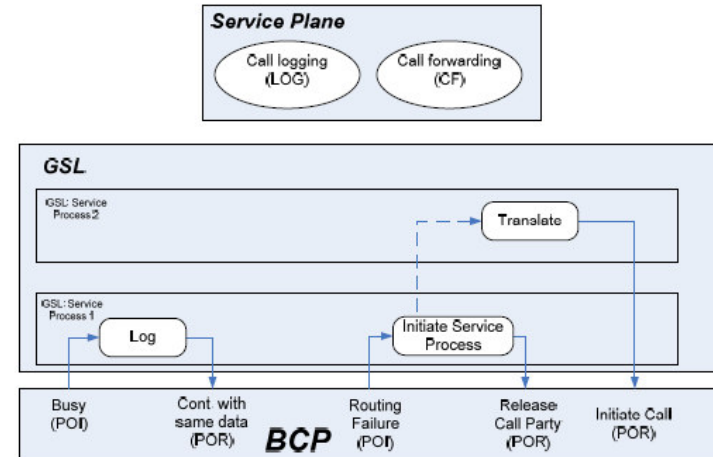
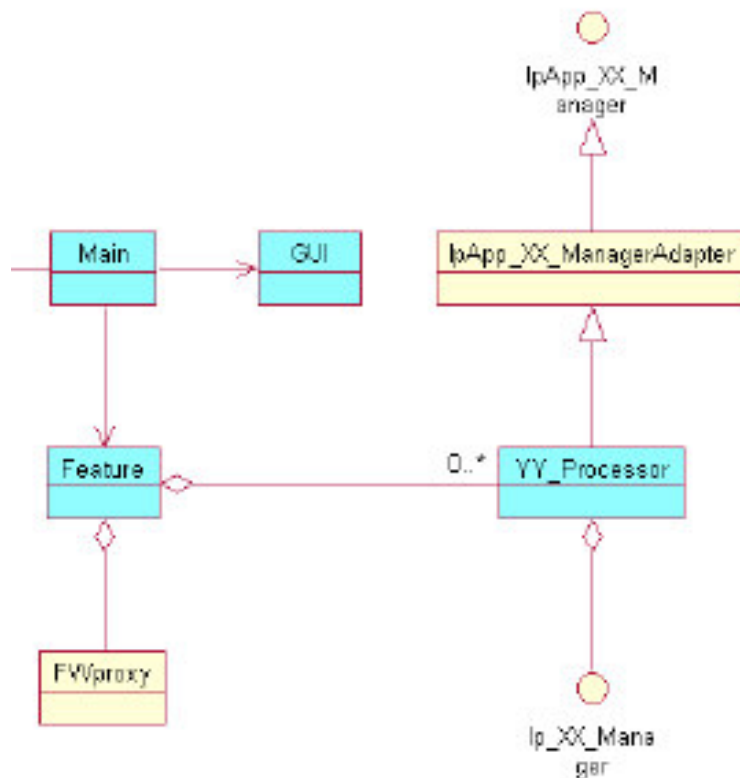
- Estudar e verificar experimentalmente:
  - O ciclo de vida de uma aplicação (provisão, aprovisionamento)
  - Interação da Aplicação com a Plataforma (SCS)
    - Utilização de SCS
    - Processo de “armar DP’s”
    - Envio de notificações do SCS para a Aplicação





# 1.3. Aplicação CallRedirect (23)

- Estudar e verificar experimentalmente:
  - O ciclo de vida de uma aplicação (provisão, aprovisionamento)
  - Interação da Aplicação com a Plataforma (SCS)



# Lab.01: CallRedirect Lógica-ROUTING\_FAILURE (UML Diagrama de Sequência)

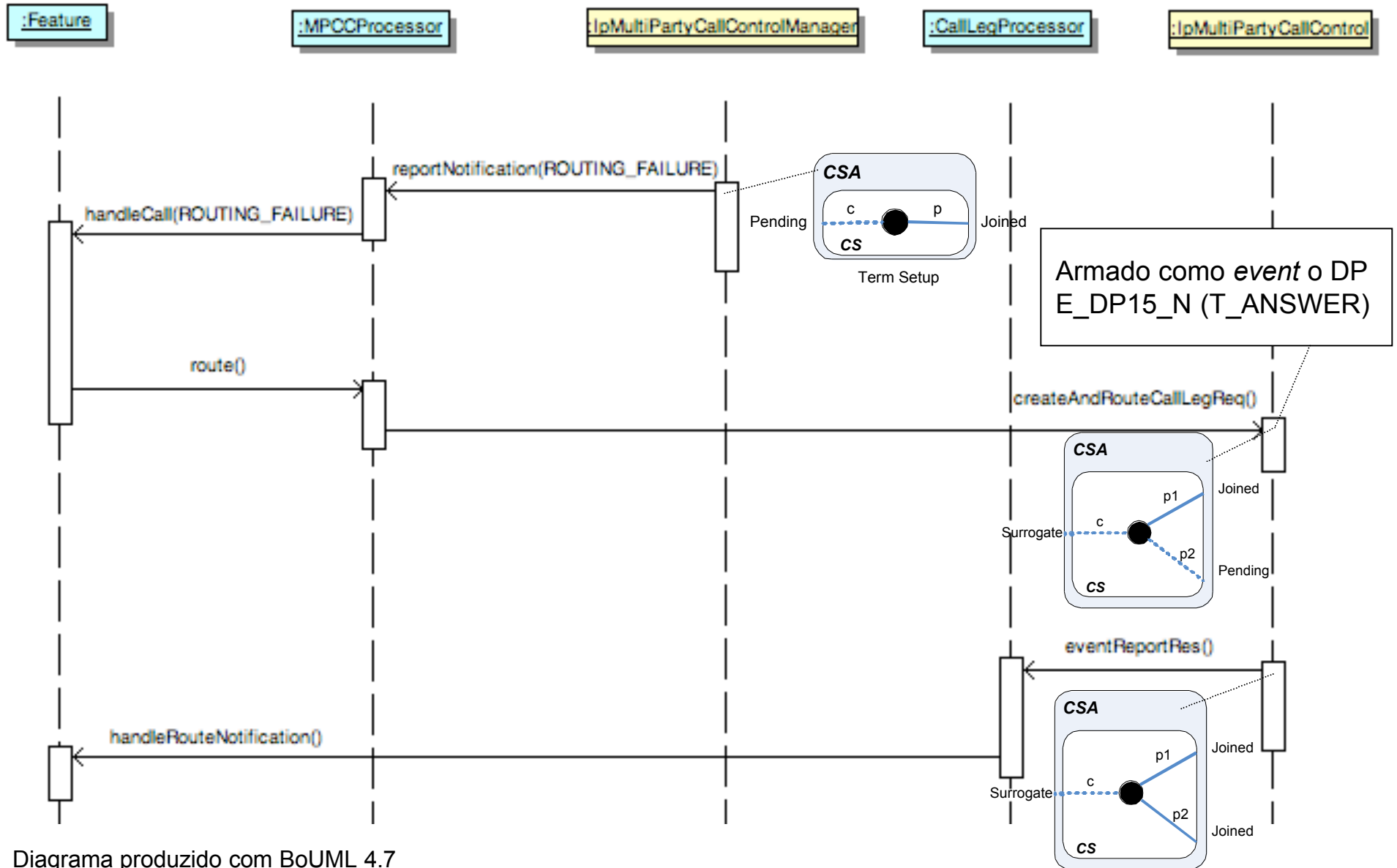


Diagrama produzido com BoUML 4.7